**基于webpack3实现的全面综合型多功能vue项目构建框架**

**本框架是一个包含框架页以及下属模块页，可选配，可按需命令行指定打包，也可根据配置项打包，每个单页面可以合并打包，也可以分开打包，可以指定部分模块启动的的多功能vue项目构建脚手架。**

**此构建框架主要功能点有：**

1. **支持构建vue单项目单页面应用；**
2. **支持构建vue单项目多页面应用；**
3. **支持单项目多页面公共资源统一打包；**
4. **支持单项目多页面各页面模块独立打包；**
5. **支持命令行指定模块整体打包；**
6. **支持命令行指定模块分开打包；**
7. **支持依据配置文件所配模块整体打包或分开打包；**
8. **包含框架页（可自动或根据参数设置实现框架页加载链接并实现跳转下属单页面模块）；**
9. **支持自动读取并打包所有下属模块以及框架页模块；**
10. **支持开发启动就可预览最终效果；**
11. **支持因多页面启动编译服务太重而为减少编译时间更改配置项从而只启动配置指定模块。**
12. **构建框架已预安装和配置常用功能模块；**
13. **支持框架页下属模块可插可拔。**

**采用本构建框架，开发者无需使用其他vue框架(vue-cli2,vue-cli3)，无需熟悉webpack技术，无需查看构建配置，就可以快速上手，并用其开发和构建一个单页面应用项目或一个可分开和整体打包的或某些模块整体打包某些模块分开打包的多页面应用项目，该框架真正意义上实现了一架多构（一个vue构建框架可以构建所有的vue项目），一切皆可配和高度灵活性。**

**实现逻辑**

**传统的vue-cli2和vue-cli3构建框架默认只支持构建单页面项目应用，不支持构建多页面应用，且开发者只能后期查阅webpack技术和vue-cli3配置项选配多页面应用，构建形式单一，即使是多页公共资源多以整体打包为主，而vue-cli2只支持单页面构建（build）,于是乎我们何不在分开构建时多次调用该build命令实现多次构建呢。打包的每一个模块名字以单页面模块命名命，子文件包含inndex.html，static静态文件。传统Build默认支持构建单页面，多页面需要配置构建配置且仅仅支持以数组的形式配置多个入口进行打包，默认不支持自动读取项目的所有的单页面模块自动实现多入口打包。**

**Webpack打包vue项目默认从每个单页面的入口js开始构建，最终将打包后的js,css插入到模版html中。于是我们可以通过读取文件夹和通配符来自动读取和匹配各个单页面模块，来循环进行打包构建。**

**npm run build 模块1,模块1 totle(第二个参数默认total,可以不传入该参数)整体打包。**

**npm run build 模块名 separate可以独立构建参数指定的单一模块。**

**npm run build-all 打包所有模块，该命令脚本通过循环调用npm run build 模块名 separate来进行分开打包。**

**我们可以通过npm run build-all命令和npm run build命令来进行分开打包（打包所有模块，然后每个模块彼此独立，每个单页面为dist文件夹下一个独立的文件夹，子文件包含index.html和static文件夹，有几个模块，就产生几个静态资源包，这种方式不会复用重复的资源，）和整体打包（打包全部模块到dist资源包下面，每个模块的入口是模块名.html文件，静态资源都在static目录中，这种方式可以复用重复的资源）。npm run build-all 和 npm run build后面也可以附加以逗号隔开的模快名或配置项指定的模块来进行打包，大大扩展了打包的灵活性，框架页可以通过js手段实现自动构建框架链接下属各单页面模块，从而实现可多可少的自动化生产。**

**具体实现细节技术参考项目构建文件。**

**使用说明**

**因该构建框架高度灵活性，所以运行命令相比传统vue-cli2或vue-cli3较多，但都较容易理解和上手。projectConfig/index为项目配置，若要更改默认的src下modules子文件夹名称或src下framePage文件夹名称，必须配置此js文件。当isbuildassignmodules为真时，项目打包或启动只会构建此配置buildmodules指定的包,命令行所传参数无效,当前状态下只可使用npm run build或npm run build-all 打包或npm run dev启动项目。当buildmodules不包含框架页面，此时npm run dev启动项目后系统默认打开页面为buildmodules[0]模块。当buildmodules包含框架页面，此时npm run dev启动项目后系统默认打开页面为框架页面。**

**以下为所有可执行命令，其中”$:”后为可执行命令**

**install dependencies 安装依赖**

**$: npm i**

**serve with hot reload at localhost:8080 运行，开发模式命令行暂不可指定运行模块，可配置rojectConfig/index来指定**

**$:npm run dev //开发模式启动**

**build for production with minification //项目构建**

**$:npm run build**

**打包全部模块到一个资源包下面，每个模块的入口是module.html文件，静态资源都在static目录中，这种方式可以复用重复的资源**

**$:npm run build moduleName1,moduleName2,...**

**打包指定模块到一个资源包下面,每个模块的入口是module.html文件，静态资源都在static目录中，这种方式可以复用重复的资源**

**$:npm run build moduleName1,moduleName2,... total**

**total代表整体打包（静态资源在同一个目录下，可以复用重复的文件），separate代表分开打包（静态资源按模块名称分别独立打包，不能复用重复的文件）默认第二个参数为total 。**

**separate模式下一次只能打包一个单页面应用，例如:npm run build a separate，该命令被npm run build-all循环调用**

**$:npm run build-all**

**打包所有模块，然后每个模块彼此独立，有几个模块，就产生几个静态资源包，这种方式不会复用重复的资源,该命令依次循环调用了npm run build 模块名 separate**

**$:npm run build-all moduleName1,moduleName2,...**

**打包制定模块，参数逗号隔开，然后每个模块彼此独立，有几个模块，就产生几个静态资源包，这种方式不会复用重复的资源**

**$:npm run build --report**

**build for production and view the bundle analyzer report**

**构建并通过查看每个js包分析报告**

**$: npm run unit**

**run unit tests**

**运行单元测试**

**$: npm run e2e**

**run e2e tests**

**运行e2e测试**

**$ npm test**

**run all tests**

**运行所有测试**

**备注：**

**（1）npm run build //总体打包时访问参数传递为framePage.html?manulist=a,b&type=totle 两个参数**

**（2）npm run build-all //访问参数参数为framePage/index.html?manulist=a,b 单个参数**

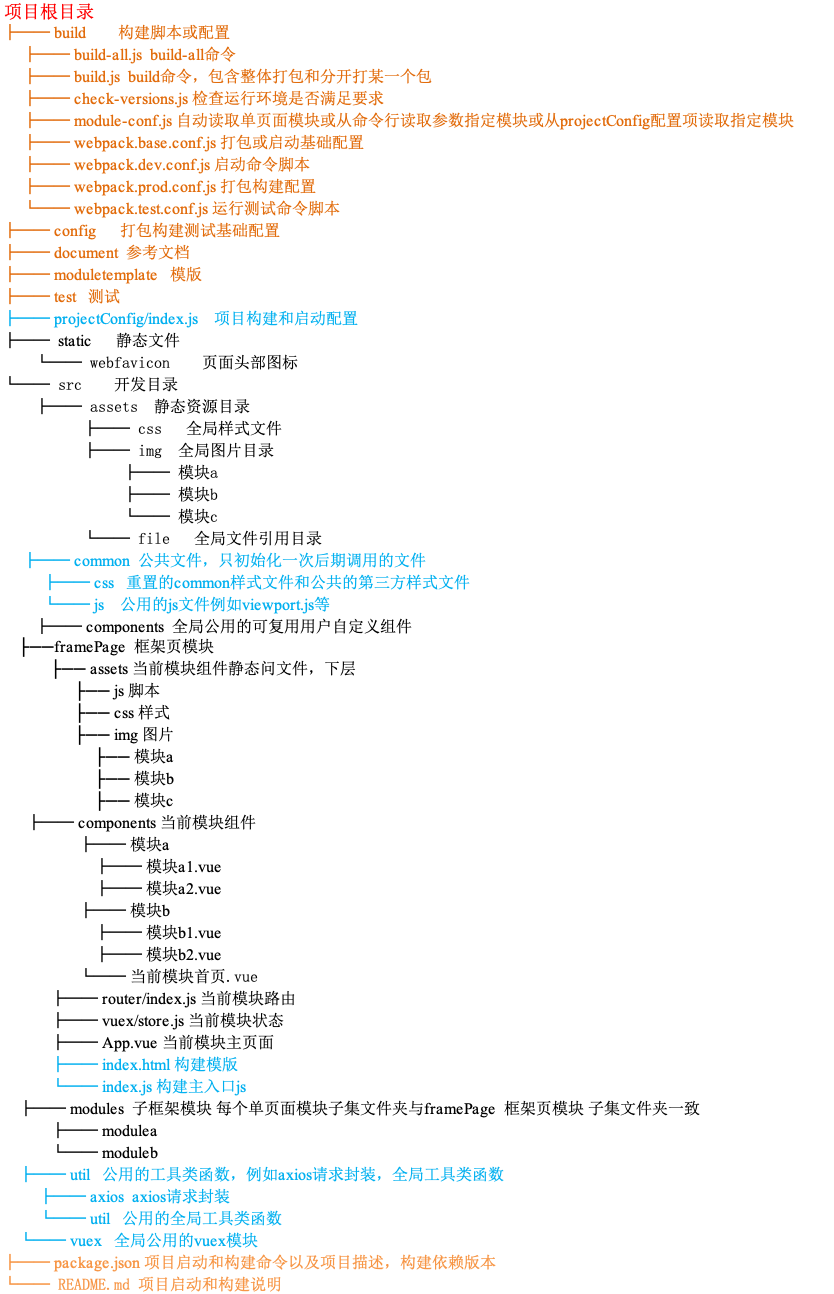
**（3）每次build时候会自动删除dist/static以及dist/\*.html 打包构建前若不想保留以前的独立打包文件时请手动删除dist文件夹。**

**附图说明**

**备注：橙色标识文件为开发者无权更改的文件目录；**

**蓝色标识文件为公共配置文件以及初始化项目初始化引入后后期不需更改的文件；**

**黑色标识文件为开发过程新建的文件以及目录模块。**

****

**图1(项目文件目录图)**

基于webpack3实现的全面综合型多功能vue项目构建框架